

INOVA  
11311 STEMMONS FRWY, SUITE 7, DALLAS, TEXAS 75229

MKR - for the Osborne Executive(tm)  
Version 1.0 - March, 1985  
by Martin Murray

## INTRODUCTION

MKR allows the owner of an Osborne Executive(tm) to attach, remove, or edit his function keys on a program (.COM) file. When the program (with attached function keys) is executed, the new function keys are automatically loaded and the program then runs normally, as if no function keys were attached. MKR also allows the creation of "NULL" files, which merely load a new set of function keys and return you to the CP/M Plus(tm) prompt. The Executive owner can optionally display the function keys as the program is loaded.

\*\*\*\*\*

The folks at INOVA prefer working with microcomputers over almost anything else. We believe in producing quality programs, as evidenced by the programs we have already written for the Osborne I and Executive, and others, and placed in the public domain. We think there are still unaddressed software needs among owners of CP/M microcomputers, and want to address those needs.

In order to continue our programming efforts, we really need your support - both as to any types of programs you wish considered, and adequate yet reasonable reward for our efforts. Please encourage your friends and associates to purchase a copy of our program for themselves, if they want it, rather than your giving or selling copies of this program to them.

\*\*\*\*\*

THIS SOFTWARE AND MANUAL ARE SOLD "AS IS" AND WITHOUT  
WARRANTIES AS TO PERFORMANCE, MERCHANTABILITY, OR FITNESS  
FOR A PARTICULAR PURPOSE.

Copyright (C) 1985 INOVA - All Rights Reserved

Sale of this software conveys a license for its use on a single computer owned or operated by only the purchaser. Copying this software or documentation by any means whatsoever for any other purpose is strictly prohibited. CP/M Plus is a registered trademark of Digital Research, Inc. WordStar is a registered trademark of MicroPro, International. Osborne Executive is a registered trademark of Osborne Computer Corporation.



## TABLE OF CONTENTS

INTRODUCTION . . . . .	1
INSTALLATION . . . . .	3
NOTES . . . . .	3
USING MKR . . . . .	3
The Menu . . . . .	4
Loading Function Keys from Memory . . . . .	4
Attaching Function Keys to a COM File . . . . .	4
Attaching Function Keys to a NULL File . . . . .	6
Loading Function Keys from a File . . . . .	7
Editing Function Keys	
in Memory . . . . .	7
in a COM or NULL File . . . . .	9
Removing Function Keys from a COM File . . . . .	9
Possible Error Messages . . . . .	10
UPDATES, LICENSES, WARRANTIES . . . . .	11



## INSTALLATION

MKR requires no installation on any standard Executive computer. It will run on any modified Executive where the use and programming of the standard function keys has not been mucked up or changed. (Trantor/GARD Exec-T owners, see note below). MKR is designed to be simple and easy to use. We request that you make a backup of the supplied floppy diskette, and put the original in a safe place.

## NOTES

In this document, a "FILENAME" (or "file name") is the first eight letters of a program name, as explained in the CP/M Plus manuals that came with the Executive. A program name will always have a filetype (the last three letters) of COM. (This is sometimes referred to as an "extension".) Since program files (or COM files) are the only ones MKR will patch, an extension of COM is always assumed.

A "NULL" file refers to a 1K program created by MKR whose sole purpose is to load into the Executive the function key definitions that it contains. After you run a "NULL" program, you will be "returned" to the CP/M prompt. This program can have any filename allowed by CP/M, but the filetype will always be COM. Example: MYKEYS.COM

Yes/No questions require a "Y" or an "N", upper or lower case. Questions asking for a filename can be aborted by simply pressing the return key at the beginning of the line.

Certain software programs, notably the pre-1985 versions of the Trantor hard disk software utilities, defeat the purpose of the arrow-key function keys by "patching" the keytables in the Executive computer directly, instead of patching the appropriate spots in the function-key definitions. In this software, the arrow-key codes are never generated -- the CP/M or WordStar control codes are produced directly. If you use this kind of software, keep in mind that only the control-number function keys (control-0 through control-9) will appear to be changed by MKR-patched programs.

## USING MKR

MKR is loaded by just typing d:MKR<cr> at the CP/M Plus prompt, where d: is the drive containing a diskette with MKR.COM on it. Our copyright message will be displayed with a menu of options. From this menu, you can perform all the operations of MKR -- we even give you a way out of the program!



### The Menu

When loaded, MKR will display a menu similar to this one:

```
MKR          (C) Copyright 1985 INOVA
Ver. 1.0     All Rights Reserved
S/N MKR010001
```

Press:	To:
1	Load function keys from memory
2	Load function keys from a COM or NULL file
3	Attach function keys to a COM file
4	Attach function keys to a NULL file
5	Remove function keys from a COM file
6	Edit function keys in a COM or NULL file
7	Edit/Display the function key set
8	Exit MKR

Enter choice: \_

### Loading Function Keys from Memory

Your current set of function keys are automatically loaded into the special MKR buffer when you first run MKR. With selection #1, you can always obtain from memory a fresh copy of the current set of function keys. When the "1" key is pressed, the menu will "flash" as the function keys are read.

### Attaching Function Keys to a COM File

(For you purists, we will discuss selection #2: "Loading Function Keys from a COM or NULL File" after we have attached a set.)

By selecting #3 from the menu, you can attach the function key definitions in the MKR buffer to a COM file. You will be given the following prompt:

Attach keys to what program: \_

Type in a program file name. You can put a "drive specifier" in front of the filename if the file exists on a different drive, and you can even put in a user number, if you are "into" that sort of thing (e.g., you are a hard-disk user). A "COM" filetype is always assumed, even if you type in something different.



Examples of valid filenames:

```
doitall
b:doitall
a15:doitall
doitallnow    <--- only first eight letters accepted
```

MKR will look for the specified file on the specified drive, and if you have forgotten the name of the file (or forgotten to put a diskette in the drive), you will see something like this:

Attach keys to what program: diddly-squat

I can't find the file. If you like, you can insert  
another disk and I will try again. Shall we do that? (y/n) \_

In this case, put the appropriate diskette in the drive before you type "Y", or type an "N" to return to the menu.

Or, if the file you specified actually exists but has been marked read-only, you will see, appropriately enough:

The file is READ ONLY. Modify anyway? (y/n) \_

(If it has been marked read-only, and YOU weren't the one who marked it that way, we recommend that you have a backup of this program before you say yes to this question.)

If you've cheated and have already attached a set of function keys to the file before you read this manual, you will be told:

File already has keys installed.  
Choose another? (y/n) \_

(If you had read the manual, you would have known that you have to "remove" the attached definitions first!)

Saying no will return you to the menu. A "Y" will ask the original question again.

If all is "cool" (the file exists and you have passed the previous tests), MKR will then prompt you with the following:

Do you wish to have the keys displayed as they are loaded? (y/n) \_



If you say yes (that is, if you type a "Y" or a "y"), MKR will include a special routine that will display the function keys as it loads them, before it runs the COM file. If you type an "N", the function keys will be loaded invisibly when the COM file is run. In either case, MKR will proceed to patch the specified COM file.

At this point, if nothing went amiss, you will be returned to the menu. MKR is actually smart enough to check the diskette and insure that there is enough room to write the modified file back to diskette, before doing so, since it is possible that the new COM file will be 1K longer than the old. (How many programs have you used that write part of a file to diskette first, THEN cry "Disk Full!" -- after the damage has been done?)

#### Attaching Function Keys to a NULL File

The #4 selection performs basically the same function as the #3 selection, except that the function keys are attached to a "nothing" file -- that is, one that is created just to load in function key definitions. Naturally, the prompt is different:

Enter the name of the new program file: \_

The disk you specify (or the "default" or logged diskette if you don't specify one) is then searched for a file of the same name. If one exists, you will then see:

File exists-delete it? (y/n) \_

If you answer "N", you will be asked:

Do you want to choose another name? (y/n) \_

A "Y" will ask the first question again; an "N" will return you to the menu.

If you pass all THESE tests, you will still have to answer whether you wish to have the function keys displayed when they are loaded. Only then will a new file be created on the diskette which can then be run all by itself to change your function keys.



### Loading Function Keys from a File

Menu item #2 allows you to load a set of function keys that have ALREADY been attached to a COM or NULL file. Selecting #2 results in:

Enter the program file name: \_

and if you specify a file that does not have any function keys previously attached by MKR, you'll see:

That file does not have any keys installed! Try another? (y/n) \_

MKR will return to the original question or the menu, depending on your answer.

After loading in a set of function keys from a file, MKR simply returns to the menu, allowing you to select a file for attaching the newly-loaded set, or to edit the newly-loaded set.

### Editing Function Keys in Memory

Keep in mind that there are two possible sets of function keys in memory, the Executive's current memory set (which are the ones loaded into the MKR buffer when MKR is first run) and the ones in the MKR buffer, which WILL BE DIFFERENT if you have loaded any sets from files or done any editing on a set.

If you wish to edit the current memory set, select #1 at the menu, then #7, "Edit/Display the function key set". If you have just loaded in a set attached to a file, or if you have already edited a set, and wish to modify it again, DO NOT select #1, as that will overwrite the set in the MKR buffer.

When you select #7, the function keys and the corresponding definitions assigned to them will be displayed, with the following prompt at the bottom of the screen:

Press KEY NAME, ARROW KEYS to move cursor, RET to edit key, ESC to exit

Control characters are displayed as a single underlined character; i.e., control-M (carriage return) is displayed as M, and ESC is displayed as [. This also applies if you choose to display function keys as they are loaded (see Attaching Function Keys, above).



You can now edit the function key set, or exit by pressing the escape key. To select a key to edit, either press a KEY NAME (a number from 0 to 9, or one of the letters U, R, D, or L) or use the up/down arrows to position the cursor in front of the key desired, then press RETURN.

You'll note that the arrow keys are indicated by U for up, R for right, D for down, and L for left. The arrow keys can be individually programmed in exactly the same manner as the other function keys. Here are the standard CP/M and WordStar definitions of the arrow keys, as well as our Special configuration -- a combination of both -- that works with WordStar and CP/M and all but a very few programs:

<u>Arrow</u>	<u>CP/M</u>	<u>WordStar</u>	<u>Special</u>
up	ctrl-K	ctrl-E	ctrl-E
right	ctrl-L	ctrl-D	ctrl-D
down	ctrl-J	ctrl-X	ctrl-X
left	ctrl-H	ctrl-S	ctrl-H

When you have selected a key, the bottom line of the screen will show:

^ESC=abort,UP=insert ON ,DOWN=del,LEFT-RIGHT move cursor,^RET=done 228

and you will be positioned on the first character of the function key. Type a control-escape to quit editing the key at any time without saving any changes you have made.

The up-arrow toggles the insert on/off mode. When insert is on, keys that you press will be "inserted" before the character at the cursor, and all characters from the cursor to the end of the line will move to the right -- unless there is no more room for entering characters. (You will hear a nasty beep when you run out of room or type an inappropriate key, like a left arrow at the beginning of the line.) When insert is off, characters you type will "overwrite" any existing characters.

The number of character spaces remaining to be programmed is displayed at the bottom right of the screen.

Use the left- and right- arrow keys to move the cursor for editing. The down-arrow key will delete the character at the cursor, if there is one, and all characters to the right of the cursor will move left.

To save your new definition, type control-RETURN. This will position the cursor on the KEY NAME of the key you just edited. You can then edit another key or press ESC to exit.

When you exit from editing the function keys, you will be returned to the menu.



### Editing Function Keys in a COM or NULL file

By selecting menu item #6, "Edit function keys in a COM or NULL file", you will be asked to enter the program file name:

Enter the program file name: \_

If the file you selected was marked read-only, you will see:

The file is READ ONLY. Modify anyway? (y/n) \_

If you indicate "No" by pressing an "N", you will be returned to the menu.

After finding a valid set of function keys attached to the specified file, MKR will enter the editing mode, which is identical with function #7 as described above. The difference between function #6 and function #7 is in entering and exiting the edit mode -- function #7 just edits what is in the MKR buffer, while function #6 will read in a function key set attached to a file, allow you to edit it, and THEN re-write the edited function keys back out to the file from whence they came. A successful re-installation will return you to the menu.

Selection #6, therefore, is the combination of #2, #7, and #3 or #4, providing you with a very quick method of updating the function keys you have previously attached to a COM or NULL file.

### Removing Function Keys from a COM File

If you choose #5 from the menu choices, MKR will display:

Enter the program file name: \_

Look familiar? You'll get the same error messages for the same errors as before. If you do all the right things, MKR will remove the function keys from the specified COM file. You can even do a "wrong" thing: You can tell MKR to remove function keys from a NULL file, and MKR will oblige, leaving you with a program that occupies 1K bytes of space on your disk and does nothing but a warm boot when run. (We suggest that you erase NULL files that you do not want.)



### Possible Error Messages

You may see various error messages if something went royally wrong. For example, running MKR on a computer other than an Executive would result in an error message: "EXECUTIVES ONLY!" or some such expletive.

If you try to add function keys to a zero-length file, you'll see a message regarding the inability of MKR to attach keys to an empty file.

If your diskette is write-protected, MKR will say, "For some reason I cannot make the file. Should I try again?" Check your diskette if you see this message.

Other messages result from pulling out the diskette in the middle of a disk write (We do not recommend that you try to find out what that message is!) or changing diskettes after reading in a file but before writing the same file back out.